

Hausaufgabe 5

Erklärung der Berechnung von Landtagswahlergebnissen in SQL

Allgemeine Vermerke

- Alle Berechnungen werden natürlich getrennt für jede Wahl vollzogen
- Mandate erhalten nur diejenigen Parteien, welche mindestens 5% der Gesamtstimmen über alle Regierungsbezirke hinweg bekommen haben
- Abgesehen von der Sperrklausel geschieht jede weitere Berechnung getrennt für jeden der sieben Regierungsbezirke

Berechnung der Listenreihenfolge der Kandidaten

```
-- Summe aller Stimmen die ein kandidat erhalten hat pro regierungsbezirk und wahl.
WITH gesamtstimmen_pro_kandidat (wahl_id, regierungsbezirk_id, kandidat_id, anzahl) AS (
  SELECT kgs.wahl_id, sk.regierungsbezirk_id, kgs.kandidat_id, sum(kgs.anzahl)
  FROM "landtagswahlen".kandidatgebundene_gueltige_stimmen kgs
  JOIN "landtagswahlen".stimmkreise sk ON sk.id = kgs.stimmkreis_id
  GROUP BY kgs.wahl_id, sk.regierungsbezirk_id, kgs.kandidat_id
)
SELECT gpk.wahl_id, gpk.regierungsbezirk_id, k.id, row_number() over (
  PARTITION BY gpk.wahl_id, gpk.regierungsbezirk_id, k.partei_id
  ORDER BY gpk.anzahl DESC
) as finalerListenplatz
FROM gesamtstimmen_pro_kandidat gpk JOIN "landtagswahlen".kandidaten k ON gpk.kandidat_id = k.id
```

Zunächst werden die erhaltenen Erst- und Zweitstimmen von jedem Kandidaten pro Regierungsbezirk aufsummiert in *gesamtstimmen_pro_kandidat*. Nachfolgend wird die *row_number()* Funktion genutzt um den Listenplatz als Zahl zu bestimmen. Hierzu werden die Gesamtstimmen nach Wahl, Regierungsbezirk und Partei partitioniert und absteigend nach Stimmenzahl sortiert. Die Partitionierung erfolgt weil es pro Wahl, Regierungsbezirk und Partei je eine Liste gibt.

Berechnung der Gesamtstimmen pro Partei

```
-- Anzahl kandidatengebundener erst+zweitstimmen pro partei
WITH kandidatgebundene_gesamtstimmen_pro_partei (wahl_id, regierungsbezirk_id, partei_id, anzahl) AS (
  SELECT kgs.wahl_id, sk.regierungsbezirk_id, k.partei_id, sum(kgs.anzahl)
  FROM "landtagswahlen".kandidatgebundene_gueltige_stimmen kgs
  JOIN "landtagswahlen".stimmkreise sk ON sk.id = kgs.stimmkreis_id
  JOIN "landtagswahlen".kandidaten k ON k.id = kgs.kandidat_id
  GROUP BY kgs.wahl_id, sk.regierungsbezirk_id, k.partei_id
),
-- Anzahl stimmen für die liste/den Wahlvorschlag pro partei
listengebundene_gesamtstimmen_pro_partei (wahl_id, regierungsbezirk_id, partei_id, anzahl) AS (
  SELECT lgs.wahl_id, sk.regierungsbezirk_id, lgs.partei_id, sum(lgs.anzahl)
  FROM "landtagswahlen".listengebundene_gueltige_stimmen lgs
  JOIN "landtagswahlen".stimmkreise sk ON sk.id = lgs.stimmkreis_id
  GROUP BY lgs.wahl_id, sk.regierungsbezirk_id, lgs.partei_id
)
SELECT kggs.wahl_id, kggs.regierungsbezirk_id, kggs.partei_id, kggs.anzahl + lggs.anzahl as anzahl
FROM kandidatgebundene_gesamtstimmen_pro_partei kggs
JOIN listengebundene_gesamtstimmen_pro_partei lggs
  ON lggs.regierungsbezirk_id = kggs.regierungsbezirk_id AND lggs.partei_id = kggs.partei_id AND
  lggs.wahl_id = kggs.wahl_id
```

Für nachfolgende Berechnungen ist es notwendig die Gesamtstimmen die jede Partei pro Regierungsbezirk erhalten hat zu kennen. Hierzu werden die Erst- und Zweitstimmen, welche Kandidaten der Partei erhalten haben (*kandidatgebundene_gesamtstimmen_pro_partei*), zu den Listengebundenen Stimmen (*listengebundene_gesamtstimmen_pro_partei*), i.e., Kreuze neben der Partei auf dem Zweitstimmzettel, aufaddiert. Es handelt sich bei beiden um getrennte Relationen in unserem Schema.

Berechnung der Mandate die jeder Partei zustehen

```
-- Anzahl der Mandate, die einer Partei pro Regierungsbezirk zustehenden (Hare-Niemeyer)
CREATE OR REPLACE FUNCTION zustehende_mandate(p_ausgleichsmandat_anzahl integer)
RETURNS TABLE (
    wahl_id smallint,
    regierungsbezirk_id smallint,
    partei_id smallint,
    anzahl int
)
AS $$
-- Gesamtstimmen für jeden Regierungsbezirk
WITH gesamttimmen (wahl_id, regierungsbezirk_id, anzahl) AS (
    SELECT wahl_id, regierungsbezirk_id, sum(anzahl)
    FROM "landtagswahlen".gesamttimmen_pro_partei
    GROUP BY wahl_id, regierungsbezirk_id
),
-- Die Parteien, welche nicht gesperrt sind für die Wahl
nicht_gesperrte_parteien (wahl_id, partei_id) AS (
    SELECT gspp.wahl_id, gspp.partei_id
    FROM (
        SELECT wahl_id, partei_id, sum(anzahl) as anzahl
        FROM "landtagswahlen".gesamttimmen_pro_partei
        GROUP BY wahl_id, partei_id
    ) gspp
    -- Gesamtstimmen gruppiert für Sperrklausel. Hier zählen
    -- alle regierungsbezirke und nicht nur der momentane
    JOIN (
        SELECT wahl_id, sum(anzahl) as anzahl
        FROM "landtagswahlen".gesamttimmen_pro_partei
        GROUP BY wahl_id
    ) gs ON gs.wahl_id = gspp.wahl_id
    WHERE gspp.anzahl / gs.anzahl >= 0.05
),
-- Anzahl der regulären mandate (direkt + liste) die in einem
-- Regierungsbezirk zu vergeben sind
gesamtmandat_anzahl (regierungsbezirk_id, wahl_id, anzahl) AS (
    SELECT dm.regierungsbezirk_id,
        dm.wahl_id,
        dm.anzahl + rwi.anzahllistenmandate + p_ausgleichsmandat_anzahl as anzahl
    FROM "landtagswahlen".direktmandat_anzahl dm
    JOIN "landtagswahlen".regierungsbezirk_wahlinfo rwi
    ON rwi.regierungsbezirk_id = dm.regierungsbezirk_id AND rwi.wahl_id = dm.wahl_id
),
```

Die Berechnung der Mandate, welche jeder Partei pro Regierungsbezirk nach Hare-Niemeyer Verfahren zustehen wird als SQL Function abgebildet, um diese in Abhängigkeit vom freien Parameter *p_ausgleichsmandat_anzahl* durchführen zu

können. Da Parteien im Bayerischen Landtagswahlverfahren mehr Sitze durch Direktmandate gewinnen können als ihnen prozentual anhand der im Regierungsbezirk erhaltenen Stimmen gerechnet zustehen würden kann es sein, dass Ausgleichsmandate an andere Parteien vergeben werden. Die Anzahl der Ausgleichsmandate ergibt sich, in dem man sukzessive weitere Mandate zu den zu vergebenden im Regierungsbezirk hinzuaddiert bis die durch das Hare-Niemeyer Verfahren bestimmte Anzahl an zustehenden Mandaten für alle Parteien ausreicht, d.h. keine Partei mehr Sitze im Landtag inne hat als ihr nach der Berechnung zustehen würden.

Relation	Beschreibung
<i>gesamtstimmen</i>	Anzahl gültige Stimmen pro Regierungsbezirk
<i>nicht_gespernte_parteien</i>	Jede Partei pro Wahl welche nicht an der 5% Hürde gescheitert ist. Die Hürde bezieht sich hierbei auf den Anteil der Stimmen für die Partei über alle 7 Regierungsbezirke hinweg
<i>gesamtmandat_anzahl</i>	Anzahl der zu vergebenden Mandate (Direkt, Liste) pro Regierungsbezirk. Hier wird der Eingehens erwähnte Parameter p_ausgleichsmandat_anzahl mit einbezogen
<i>sperrklausel_stimmen</i>	Aufsummiert die Anzahl der Stimmen, welche aufgrund der Sperrklausel im weiteren ignoriert werden müssen. Ergibt sich als die summe der Stimmen pro Partei für Parteien die nicht nicht gesperrt sind
<i>bereinigte_gesamtstimmen</i>	Pro Regierungsbezirk die aufsummierte Anzahl aller abgegebenen, gültigen Stimmen die nicht durch die Sperrklausel wegfallen

```
-- Anzahl der Stimmen pro Regierungsbezirk die wegen Sperrklausel wegfallen
sperrklausel_stimmen (wahl_id, regierungsbezirk_id, anzahl) AS (
  SELECT gspp.wahl_id, gspp.regierungsbezirk_id, sum(gspp.anzahl) as anzahl
  FROM "landtagswahlen".gesamtstimmen_pro_partei gspp
  WHERE NOT EXISTS (
    SELECT *
    FROM nicht_gespernte_parteien ngp
    WHERE gspp.wahl_id = ngp.wahl_id
      AND gspp.partei_id = ngp.partei_id
  )
  GROUP BY gspp.wahl_id, gspp.regierungsbezirk_id
),
-- Gesamtstimmen pro Regierungsbezirk exclusive Sperrklausel Stimmen
bereinigte_gesamtstimmen (wahl_id, regierungsbezirk_id, anzahl) AS (
  SELECT gs.wahl_id, gs.regierungsbezirk_id, gs.anzahl - sks.anzahl
  FROM gesamtstimmen gs
  JOIN sperrklausel_stimmen sks
  ON gs.regierungsbezirk_id = sks.regierungsbezirk_id AND gs.wahl_id = sks.wahl_id
),
```

Relation	Beschreibung
<i>sitzquoten</i>	Hare-Niemeyer Quote = $\text{Gesamtsitzzahl} \cdot \frac{\text{Parteistimmenzahl}}{\text{Gesamtstimmmanzahl}}$
<i>uebrigemandate</i>	Jede Partei erhält nach Hare-Niemeyer Verfahren ihre jeweilige abgerundete Quote als Sitze direkt. Die übrigen Sitze werden gemäß der gröÙe der Abrundungsreste der Quoten (<i>gma.anzahl</i> - <i>sqq.anzahl</i>) an die Parteien vergeben, damit das Verfahren Summenerhaltend bleibt. <i>uebgrigemandate</i> ermittelt genau die Anzahl der Mandate die nach dem initialen Abrundungsschritt pro Regierungsbezirk übrig sind

-- Hare-Niemeyer quotas per regierungsbezirk and partei

sitzquoten (wahl_id, regierungsbezirk_id, partei_id, quote) AS (

```

SELECT bgs.wahl_id,
       bgs.regierungsbezirk_id,
       gspp.partei_id,
       gma.anzahl * gspp.anzahl / bgs.anzahl as quote
FROM   bereinigte_gesamtstimmen bgs
       JOIN gesamtstimmen gs
         ON gs.wahl_id = bgs.wahl_id
         AND gs.regierungsbezirk_id = bgs.regierungsbezirk_id
       JOIN "landtagswahlen".gesamtstimmen_pro_partei gspp
         ON gspp.wahl_id = bgs.wahl_id
         AND gspp.regierungsbezirk_id = bgs.regierungsbezirk_id
       JOIN nicht_gesperrte_parteien ngp
         ON gspp.wahl_id = ngp.wahl_id
         AND gspp.partei_id = ngp.partei_id
       JOIN gesamtmandat_anzahl gma
         ON gma.wahl_id = bgs.wahl_id
         AND gma.regierungsbezirk_id = bgs.regierungsbezirk_id

```

),

-- Durch abrunden bleiben Sitze übrig. Damit Verfahren summenerhaltend ist

-- werden übrige Mandate verteilt:

uebgrigemandate (wahl_id, regierungsbezirk_id, anzahl) AS (

```

SELECT sqg.wahl_id, sqg.regierungsbezirk_id, gma.anzahl - sqg.anzahl
FROM   (
  SELECT wahl_id, regierungsbezirk_id, sum(floor(quote)) as anzahl
  FROM   sitzquoten sq
  GROUP BY wahl_id, regierungsbezirk_id
) sqg
       JOIN gesamtmandat_anzahl gma
         ON gma.wahl_id = sqg.wahl_id
         AND gma.regierungsbezirk_id = sqg.regierungsbezirk_id

```

),

Relation	Beschreibung
<i>zusatzsitze</i>	Ermittelt welche Partei ein Anrecht auf Zusatzsitze hat. Da LIMIT keine Variablen nimmt wird hier wieder <i>row_number()</i> verwendet um den Parteien pro wahl und Regierungsbezirk gemäß der Größe ihres Quotenrests eine Zahl zuzuordnen, welche dann verwendet wird um nur den ersten k Parteien Zusatzmandate zu geben für k = Anzahl übriger Mandate
Äußeres SELECT	Verteilt die Zusatzsitze an die Parteien gemäß der Quoten. CASE ... 1 ELSE 0 funktioniert, da die Quote abhängig von der Sitzanzahl inklusive <i>p_ausgleichsmandat_anzahl</i> ist und $Quote - \lfloor Quote \rfloor < 1$ gilt.

```

-- Pro Wahl und Regierungsbezirk stehen hier Parteien die je einen
-- Zusatzsitz erhalten damit die Sitzverteilung summenhaltend bleibt
zusatzsitze (wahl_id, regierungsbezirk_id, partei_id) AS (
  SELECT sqr.wahl_id, sqr.regierungsbezirk_id, sqr.partei_id
  FROM (
    SELECT *,
      -- Da LIMIT keine Variablen nimmt muss
      -- man mit row_number() arbeiten
      row_number() OVER (
        PARTITION BY wahl_id, regierungsbezirk_id
        ORDER BY quote - floor(quote) DESC
      ) AS row_number
    FROM sitzquoten sq
  ) sqr
  JOIN uebgrigemandate um
  ON sqr.wahl_id = um.wahl_id
  AND sqr.regierungsbezirk_id = um.regierungsbezirk_id
  WHERE sqr.row_number <= um.anzahl
  ORDER BY sqr.wahl_id, sqr.regierungsbezirk_id
)
SELECT sq.wahl_id,
  sq.regierungsbezirk_id,
  sq.partei_id,
  CAST(floor(sq.quote) as integer) + (
    CASE
      WHEN EXISTS(
        SELECT *
        FROM zusatzsitze zs
        WHERE zs.wahl_id = sq.wahl_id
          AND zs.regierungsbezirk_id = sq.regierungsbezirk_id
          AND zs.partei_id = sq.partei_id
      )
      THEN 1
      ELSE 0 END
  )
  FROM sitzquoten sq
$$ LANGUAGE SQL
STABLE;

```

Berechnen der gewonnenen Direktmandate

Zur Berechnung der gewonnenen Direktmandate müssen zunächst alle Kandidaten gesperrter Parteien rausgefiltert werden. Ebenso dürfen nur Direktkandidaten eines Stimmkreises miteinander verglichen werden. Gewinner eines Stimmkreises ist dann derjenige Direktkandidat, für den kein anderer Direktkandidat des Stimmkreises existiert, der mehr Stimmen erhalten hat.

```
WITH gesamtstimmen (wahl_id, anzahl) AS (  
    SELECT wahl_id, sum(anzahl)  
    FROM "landtagswahlen".gesamtstimmen_pro_partei  
    GROUP BY wahl_id  
) ,  
-- Die Parteien, welche nicht gesperrt sind für die Wahl  
nicht_gesperrte_parteien (wahl_id, partei_id) AS (  
    SELECT gspp.wahl_id, gspp.partei_id  
    FROM (  
        SELECT wahl_id, partei_id, sum(anzahl) as anzahl  
        FROM "landtagswahlen".gesamtstimmen_pro_partei  
        GROUP BY wahl_id, partei_id  
    ) gspp  
    JOIN gesamtstimmen gs ON gs.wahl_id = gspp.wahl_id  
    WHERE gspp.anzahl / gs.anzahl >= 0.05  
) ,  
nicht_gesperrte_direktkandidaten (wahl_id, stimmkreis_id, kandidat_id,  
    stimmanzahl) AS (  
    SELECT dk.wahl_id, dk.stimmkreis_id, dk.direktkandidat_id, kgs.anzahl  
    FROM "landtagswahlen".direktkandidaten dk  
    JOIN "landtagswahlen".kandidaten k  
    ON k.id = dk.direktkandidat_id  
    JOIN "landtagswahlen".stimmkreise sk  
    ON sk.id = dk.stimmkreis_id  
    JOIN "landtagswahlen".kandidatgebundene_guelteige_stimmen kgs  
    ON kgs.kandidat_id = dk.direktkandidat_id  
    AND kgs.wahl_id = dk.wahl_id  
    AND kgs.stimmkreis_id = dk.stimmkreis_id  
    JOIN nicht_gesperrte_parteien ngp  
    ON ngp.wahl_id = dk.wahl_id  
    AND ngp.partei_id = k.partei_id  
)  
SELECT ngd1.wahl_id, ngd1.stimmkreis_id, ngd1.kandidat_id  
FROM nicht_gesperrte_direktkandidaten ngd1  
WHERE NOT EXISTS(  
    SELECT *  
    FROM nicht_gesperrte_direktkandidaten ngd2  
    WHERE ngd1.wahl_id = ngd2.wahl_id  
    AND ngd1.stimmkreis_id = ngd2.stimmkreis_id  
    AND ngd2.stimmanzahl > ngd1.stimmanzahl  
)
```


Berechnung der gewonnenen Listenmandate

Die Berechnung der gewonnenen Listenmandate muss rekursiv erfolgen um die Anzahl der notwendigen Ausgleichsmandate feststellen zu können. Weiterhin kommt hier die vorher definierte Funktion *zustehende_mandate(...)* zum Einsatz.

```
-- Anzahl gewonnener direktmandate pro partei und Regierungsbezirk
WITH RECURSIVE anzahl_gewonnene_direktmandate (wahl_id, regierungsbezirk_id, partei_id, anzahl) AS
  SELECT gdm.wahl_id, sk.regierungsbezirk_id, k.partei_id, count(*)
  FROM "landtagswahlen".gewonnene_direktmandate gdm
  JOIN "landtagswahlen".stimmkreise sk
  ON sk.id = gdm.stimmkreis_id
  JOIN "landtagswahlen".kandidaten k
  ON k.id = gdm.kandidat_id
  GROUP BY gdm.wahl_id, sk.regierungsbezirk_id, k.partei_id
),
-- Anzahl Listenmandate die jeder Partei zustehen
listenmandate_recursive (wahl_id, regierungsbezirk_id, partei_id, anzahl, zusatzmandate) AS (
  (
    SELECT z.wahl_id,
           z.regierungsbezirk_id,
           z.partei_id,
           z.anzahl - COALESCE(agd.anzahl, 0) as anzahl,
           0 as zusatzmandate
    FROM zustehende_mandate(0) z
    LEFT OUTER JOIN anzahl_gewonnene_direktmandate agd
    ON z.wahl_id = agd.wahl_id
    AND z.regierungsbezirk_id = agd.regierungsbezirk_id
    AND z.partei_id = agd.partei_id
  )
  UNION
  (
    SELECT z.wahl_id,
           z.regierungsbezirk_id,
           z.partei_id,
           z.anzahl - COALESCE(agd.anzahl, 0) as anzahl,
           CAST(zlm1.zusatzmandate + 1 as int) as zusatzmandate
    FROM listenmandate_recursive zlm1
    JOIN zustehende_mandate(CAST(zlm1.zusatzmandate + 1 as int)) z
    ON zlm1.wahl_id = z.wahl_id
    AND zlm1.regierungsbezirk_id = z.regierungsbezirk_id
    LEFT OUTER JOIN anzahl_gewonnene_direktmandate agd
    ON z.wahl_id = agd.wahl_id
    AND z.regierungsbezirk_id = agd.regierungsbezirk_id
    AND z.partei_id = agd.partei_id
    WHERE zlm1.anzahl < 0
  )
),
```

Relation	Beschreibung
<i>anzahl_gewonnene_direktmandate</i>	Für jede Partei wie viele Direktmandate diese gewonnen hat im Regierungsbezirk. Die Kenntnis dieser Zahl ist notwendig um zu bestimmen, auf wie viele Listenmandate eine Partei Anspruch hat.


```

-- Pro partei wie viele Mandate dieser zustehen und wie viele zusatzmandate
-- eingeführt werden mussten
listenmandate_pro_partei (wahl_id, regierungsbezirk_id, partei_id, anzahl, zusatzmandate) AS (
  SELECT z1.wahl_id, z1.regierungsbezirk_id, z1.partei_id, z1.anzahl, z1.zusatzmandate
  FROM listenmandate_recursive z1
  WHERE NOT EXISTS(
    SELECT *
    FROM listenmandate_recursive z2
    WHERE z1.wahl_id = z2.wahl_id
      AND z1.regierungsbezirk_id = z2.regierungsbezirk_id
      AND z1.partei_id = z2.partei_id
      AND z1.zusatzmandate < z2.zusatzmandate
    )
  ORDER BY z1.partei_id
)

```

Relation	Beschreibung
<i>listenmandate_recursive</i>	Grundidee: Basierend auf dem Zustand mit einer bestimmten Anzahl k an zusätzliche Mandaten (Rekursionsbasis: <i>zustehende_mandate(0)</i>), berechne in jedem Rekursionsschritt für alle Regierungsbezirke in denen mindestens eine Partei mehr Mandate bekommen hat als ihr zustehen (<i>zlm1.anzahl < 0</i>) den Zustand für k+1 zusätzliche Mandate. Die Rekursion terminiert sobald kein Regierungsbezirk mehr existiert in dem mindestens eine Partei mehr Mandate erhalten hat als ihr zustehen, i.e., sobald die Hare-Niemeyer Berechnungen für alle Parteien passen.
<i>listenmandate_pro_partei</i>	In <i>listenmandate_recursive</i> sind auch noch alle Zwischenergebnisse enthalten. Im weiteren interessiert aber nur der Endzustand vergebener Listenmandate. Diesen erhält man, in dem man diejenigen Tupel extrahiert für die kein zweites Tupel mit mehr Zusatzmandaten existiert
Äußeres SELECT	Das äußere SELECT verteilt tatsächlich die Listenmandate. Hierbei ist wichtig, dass in der Reihenfolge der ausgezählten Listenplätze vergeben wird und Kandidaten, welche ein Direktmandat gewonnen haben, übersprungen werden. <i>flf</i> enthält alle Listenkandidaten die kein Direktmandat gewonnen haben je mit ihrer Priorität, i.e., dem Platz auf der Liste nach Auszählung und angepasst an die fehlenden Direktkandidaten. Letzteres bedeutet, dass Listenkandidaten im Listenplatz aufrücken wenn Kandidaten über ihnen eliminiert wurden (e.g., weil diese Direktmandate gewonnen haben). Das Aufrücken ist wichtig um Listenplätze einfach über das Prädikat <i>flf.mandatordnung <= lpp.anzahl</i> vergeben zu können, i.e., jedem dessen aufgerückter Platz kleiner als die Anzahl der Partei zustehenden Listenmandate ist.

```

SELECT flf.wahl_id, flf.regierungsbezirk_id, k.id, lpp.zusatzmandate
FROM (
    -- recalculate row number after excluding direkt_mandat winner to be able to
    -- check with <= predicate who won a listenmandat
    SELECT fl.wahl_id, fl.regierungsbezirk_id, fl.kandidat_id, row_number() OVER (
        PARTITION BY fl.wahl_id, fl.regierungsbezirk_id, k.partei_id
        ORDER BY fl.finalerlistenplatz
    ) as mandatorordnung
    FROM "landtagswahlen".finaleliste fl
    JOIN "landtagswahlen".kandidaten k
    ON k.id = fl.kandidat_id
    -- Exclude all kandidaten that won a direct mandate
    WHERE NOT EXISTS (
        SELECT *
        FROM "landtagswahlen".gewonnene_direktmandate gdm
        JOIN "landtagswahlen".stimmkreise sk
        ON sk.id = gdm.stimmkreis_id
        WHERE fl.wahl_id = gdm.wahl_id
        AND fl.regierungsbezirk_id = sk.regierungsbezirk_id
        AND fl.kandidat_id = gdm.kandidat_id
    )
    -- finaleListe filtered
) flf
JOIN "landtagswahlen".kandidaten k
ON k.id = flf.kandidat_id
JOIN listenmandate_pro_partei lpp
ON flf.wahl_id = lpp.wahl_id
AND flf.regierungsbezirk_id = lpp.regierungsbezirk_id
AND k.partei_id = lpp.partei_id
WHERE flf.mandatorordnung <= lpp.anzahl

```